PS 2: More Unix Commands

CMPE 230 - Spring 2024

Gökçe Uludoğan

Based on the slides by Abdullatif Köksal and Rıza Özçelik, with their permissions.

Previously in CMPE 230...

- ls
- mkdir
- cd
- pwd
- cp
- ~
- touch
- cat
- mv
- rm/rmdir
- nano/vim
- clear
- head
- tail
- grep
- wc
- >, <, >>, |
- sort
- wildcards(*)
- man

File System Access Rights

- **ls -l**
  - Lists contents of the directory with long listing

```
total 24K
-rw-rw-r-- 1 cmpe230 cmpe230  79 Mar 14 11:08 hello.cpp
-rwxrwxr-x 1 cmpe230 cmpe230 17K Mar 14 11:08 hello.o
```

File permission, number of links, owner name, owner group, file size, time of last modification, file/directory name

File System Access Rights

```
total 24K
-rw-rw-r-- 1 cmpe230 cmpe230  79 Mar 14 11:08 hello.cpp
-rwxrwxr-x 1 cmpe230 cmpe230 17K Mar 14 11:08 hello.o
```

- *rwx rwx r-x*

First character: -/d = File/directory

First triplet: rwx = Permission of the **owner** (can read, write* and execute)

Second triplet: rwx = Permission of the **user group** (can read, write and execute)

Third triplet: r-x = Permission of the **others** (can read or execute but cannot write)

** write permission also allows deletion*

File System Access Rights

```
.:
total 28
-rw-rw-r-- 1 cmpe230 cmpe230    79 Mar 14 11:08 hello.cpp
-rwxrwxr-x 1 cmpe230 cmpe230 16704 Mar 14 11:08 hello.o
drwxrwxr-x 2 cmpe230 cmpe230  4096 Mar 14 11:16 project

./project:
total 0
-rw-rw-r-- 1 cmpe230 cmpe230 0 Mar 14 11:16 main.c
```

d, -	directory, file
r	Read
w	Write
x	Executable
Number of links	1
Owner name	cmpe230
Owner group	cmpe230
# kB blocks used	28
Last modification date	Mar 14, 11:08

File System Access Rights

- `-rwxrwxrwx`
  - A file that everyone can read, write (delete), and execute

- `drw-------`
  - A directory that only the owner can read and write
  - Note that the owner cannot execute the directory i.e. ***cannot cd into it!***

Changing File System Access Rights

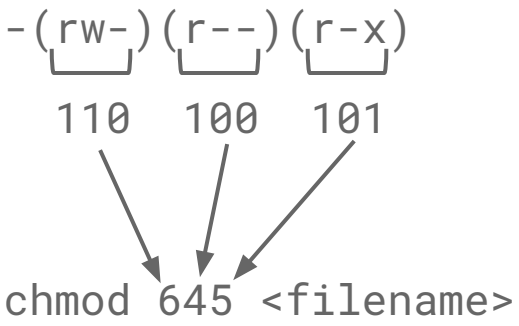
- **`chmod`**
  - Only **the owner** of a file can use `chmod` to change the permission of a file

- `chmod u+rw <filename>`
- `chmod g-rwx <filename>`
- `chmod go=r <filename>`

Symbol	Meaning
u	user (owner)
g	group
o	other
a	all
r	read
w	write/delete
x	execute
+	add permission
-	Remove permission
=	set permission

Changing File System Access Rights

- To set different permission rights for multiple user types, use binarization.
  - For instance change `-(--x)(r-x)(rwx)` to `-(rw-)(r--)(r-x)`



Changing File System Access Rights

Exercise: Write the following commands for file `hello.o`

- Remove group read/write/execute permissions
- Add executable permissions for the owner, group and other
- Allow owner and group to read and execute and block everything else
- Set permission of files and directories to `rwx------` under PS2

Changing File System Access Rights

Exercise: Write the following commands for file `hello.o`

- Remove group read/write/execute permissions
  - `chmod g-rwx hello.o`
- Add executable permissions for the owner, group and other
- Allow owner and group to read and execute and block everything else
- Set permission of files and directories to `rwx------` under PS2

Changing File System Access Rights

Exercise: Write the following commands for file `hello.o`

- Remove group read/write/execute permissions
  - `chmod g-rwx hello.o`
- Add executable permissions for the owner, group and other
  - `chmod a+x hello.o`
- Allow owner and group to read and execute and block everything else
- Set permission of files and directories to `rwx------` under PS2

Changing File System Access Rights

Exercise: Write the following commands for file `hello.o`

- Remove group read/write/execute permissions
  - `chmod g-rwx hello.o`
- Add executable permissions for the owner, group and other
  - `chmod a+x hello.o`
- Allow owner and group to read and execute and block everything else
  - (r-x)(r-x)(---)
  - 101 101 000
  - `chmod 550 hello.o`
- Set permission of files and directories to `rwx------` under PS2

Changing File System Access Rights

Exercise: Write the following commands for file `hello.o`

- Remove group read/write/execute permissions
  - `chmod g-rwx hello.o`
- Add executable permissions for the owner, group and other
  - `chmod a+x hello.o`
- Allow owner and group to read and execute and block everything else
  - (r-x)(r-x)(---)
  - 101 101 000
  - `chmod 550 hello.o`
- Set permission of files and directories to `rwx------` under PS2
  - `chmod -R 700 PS2/`

Changing File System Access Rights

- **chown**
  - Changes the owner user and the owner group of a file/folder

```
sudo useradd notme
sudo groupadd notmygroup

sudo chown notme:notmygroup myfile.txt
chmod 777 myfile.txt
```

Unix Commands

- `ps`
  - To show running processes in the current shell
  - `-e` To show every running process
  - `-f` Display in the long format
  - `-u` User oriented format, which shows %CPU, %MEM as well
  - Combine with grep to find processes of interest
- `top`
  - To show processes and CPU usage
- `kill <process_id>`
  - Terminate the process

Unix Commands

- `<any command> &`
  - & is allows non-blocking IO. In other words you can continue to use the current shell while the command is running. The process will be terminated when the terminal is closed.

- `nohup <any command>`
  - The command continues to run even if the terminal is closed.
  - Combined with &, nohup is a great tool to leave a process running in a server.

Unix Commands

- `df`
  - used to estimate file space usage
  - **-h** for human readable format

- `du <path>`
  - Calculate folder size by reading each object
  - **-a** to include files
  - **-h** for human readable format
  - **-s** for summation

Unix Commands

- `find <path> -name <pattern>`
  - Find and display files with pattern in the filepath (-delete to delete)
  - **`-perm`** find and display files with the given permission in the filepath (-delete to delete)
  - **`-exec`** to execute commands over the found files and directories
  - Link to a short tutorial

Exercise:

Allow everyone to execute .py files under the current folder

Unix Commands

- `find <path> -name <pattern>`
  - Find and display files with pattern in the filepath
  - **`-perm`** find and display files with the given permission in the filepath (-delete to delete)
  - **`-exec`** to execute commands over the found files and directories
  - Link to a short tutorial

Exercise:

Allow everyone to execute .out files under the current folder

```
find . -name *.out -exec chmod a+x {} \;
```

Unix Commands

- `mount /dev/sdb1 /mnt/media`
  - Mounts usb/cdrom to media folder.
  - Associates a file system with a particular point in the overall file system

"The **mount** command instructs the operating system that a file system is ready to use, and **associates it with a particular point** in the overall file system hierarchy (its mount point) and sets options relating to its access."

https://www.wikiwand.com/en/Mount_(Unix)

Unix Commands

- **`mount /dev/sdb1 /mnt/media`**
  - Mounts usb/cdrom to media folder

- **`umount /dev/sdb1`** OR **`umount /mnt/media`**
  - Unmounts the corresponding device (Equivalent to eject/safely remove actions in Windows)

Unix Commands

- `history`
  - Display the commands executed until now
  - Use ! followed by command order to re-execute any command
  - Use grep to search a command with keyword

- `wget/curl <link>`
  - Download the file at the link

Unix Commands

- `which <command>`
  - Identifies the location of executables

- `gzip/gunzip <filename>`
  - Zip and unzip files easily

- `diff <file1> <file2>`
  - Display differences in two files

Ubuntu Package Management

- In Ubuntu **APT**: Advanced Packaging Tool manages the packages
  - **`apt`** requires root access: **sudo**

- Installing a new package:
  - **`sudo apt-get install`** `<package_name>`

Ubuntu Package Management

- **`sudo apt-get update`**
  - Updates your source list
- **`sudo apt-get dist-upgrade`**
  - Will update **the most important packages at the expense of less important ones i**f necessary (smart dependency resolution)
- **`sudo apt-get check`**
  - Updates the package lists and checks for broken dependencies
- **`sudo apt-get autoclean`**
  - Removes .deb files (i.e. installation files) that are no longer installed in the system

Ubuntu Package Management

- `sudo apt-get remove <package_name>`
  - Removes an installed package, **leaving configuration files intact**
- `sudo apt-get purge <package_name>`
  - Completely **removes a package and the associated configuration files**
- `sudo apt-get autoremove`
  - Removes packages that were installed by other packages and are **no longer needed**

How to connect to another computer?

- `ssh username@computer`
  - Establish a secure shell connection to another computer

- `scp <localfilepath> username@computer:<remotefilepath>`
  - Send a file from one computer to another

https://confluence.atlassian.com/bamboo/generating-your-aws-private-key-file-and-certificate-file-289277123.html